Equivalence-Based Abstractions for Learning General Policies

Dominik Drexler, Simon Ståhlberg
Blai Bonet, Hector Geffner

PRL Workshop, ICAPS, 2024

Hector Geffner
RWTH Aachen University
Aachen, Germany

Linköping University
Linköping, Sweden

Motivation

- We have been looking at **two methods** for learning general policies
  - ▷ **Combinatorial:** Explicit pool of features, Min-SAT formulation
  - ▷ **GNNs:** Features learned to represent value or policy functions via DRL
- Two main issues:
  - ▷ **Scalability,** in combinatorial setting
  - ▷ **Expressivity,** in **both** settings
- Aims of this work:
  - ▷ Exploit **state symmetries** (isomorphisms) for reducing # of states in training
  - ▷ Use **symmetries** to eval expressive requirements of planning domains

Related threads

- **Symmetries in planning** for pruning search space/change problem representation [Pochter et al., 2011, Shleyfman et al., 2015, Riddle et al., 2016, Sievers et al., 2019]

- **General policies:** comb. approaches [Khardon, 1999, Martín and Geffner, 2000, Fern et al., 2006, Srivastava et al., 2008, Jiménez et al., 2019, Francès et al., 2021]; DL and DRL approaches [Toyer et al., 2020, Bajpai et al., 2018, Rivlin et al., 2020, Ståhlberg et al., 2023]

- **Expressivity:** GNNs, 1-WL, description logics, C2 [Morris et al., 2019, Barceló et al., 2020, Grohe, 2021]; in planning [Ståhlberg et al., 2024, Horcík and Šír, 2024, Drexler et al., 2024]

Planning, Generalized Planning

- A planning problem $P = \langle D, I \rangle$, **domain** D, instance info $I = \{Objs, Init, Goal\}$
- A **generalized** planning problem $\mathcal{Q}$ is set of instances P over same domain D
- A **general policy** π picks state (π) transitions (s, s') in each $P \in \mathcal{Q}$
- A general policy π **solves** P if all π-**trajectories** starting at s_0 end in goal state
- A general policy π **solves** solves $\mathcal{Q}$ if it solves all P in $\mathcal{Q}$

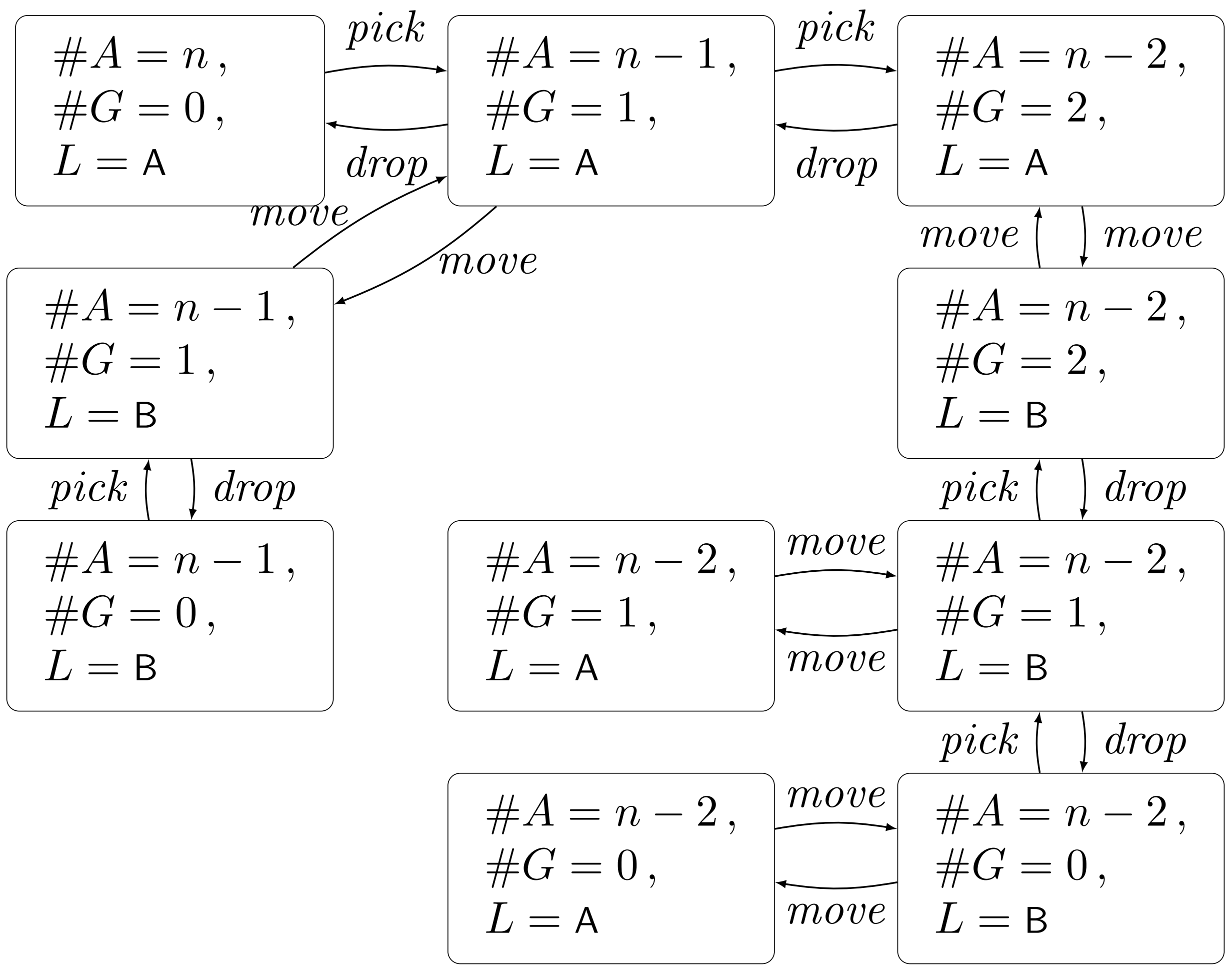
Goal atoms $p(c_1, \ldots, c_k)$ encoded in states as $p_G(c_1, \ldots, c_k)$ where p_G new predicate

Equivalence-based abstractions

- For learning π to solve $\mathcal{Q}$, look for π that solves **some** instances from $\mathcal{Q}$
- Number and size of instances required small, but if **too small**, π won't generalize
- **Idea:** prune **isomorphic states** from instances $P \in Q$
- Let $S(P) = \langle S, s_0, G, Succ \rangle$ where $(s, s') \in Succ$ if $\exists$ action that maps s into s'
- **Reduced/abstract** state model $\widetilde{S(P)} = \langle \tilde{S}, \tilde{s}_0, \tilde{G}, \widetilde{Succ} \rangle$ from $S(P)$:
  - ▷ $\tilde{S} \doteq \{[s] \mid s \in S\}$; where $[s]$ stands for class of states **isomorphic** to s
  - ▷ $[s_0]$ for initial state s_0 of P,
  - ▷ $\tilde{G} \doteq \{[s] \mid s \in G\}$ for the goal states,
  - ▷ $\widetilde{\text{Succ}} \doteq \{([s], [s']) \mid (s, s') \in \text{Succ}\}$.

Theorem: If π is a *first-order* policy, π solves $\mathcal{Q}$ over *first-order* STRIPS domain D iff π solves **set of reduced problems** $\tilde{\mathcal{Q}} = \{\widetilde{S(P)} \mid P \in Q\}$.

Example: Equivalence Reduction in Gripper



For n balls, $|S|$ is exponential in n, while $|\tilde{S}|$ is $6n$

Experiments: Gains with equivalence reduction

	with equivalence-based reduction				without equivalence-based reduction			
Domain	M	T_{pre}	T_{learn}	$\mathcal{Q}_{\mathcal{T}}/{\sim_{iso}}$	M	T_{pre}	T_{learn}	$\mathcal{Q}_{\mathcal{T}}$
Blocks3ops	9	**537**	**6,876**	**4,901**	9	992	71,299	145,680
Blocks4ops-clear	1	**2**	**3**	**86**	1	4	71	30,540
Blocks4ops-on	**2**	**60**	**185**	**249**	3	192	301	30,540
Delivery	**1**	**166**	**290**	**3,346**	3	820	15,355	411,720
Ferry	1	**14**	70	**265**	1	20	**41**	8,430
Gripper	1	4	**2**	**90**	1	**3**	7	1,084
Miconic	1	40	**37**	**17,661**	1	**10**	77	32,400
Reward	1	23	**26**	**7,026**	1	**5**	93	13,394
Spanner	1	3	**3**	**525**	1	3	5	9,291
Visitall	2	1,761	**14,163**	**446,005**	2	**46**	15,487	476,766

Learning gen policies with/without reductions; **combinatorial approach**

Memory in GiB (M), time in secs: preprocessing (T_{pre}), training/validation (T_{learn})

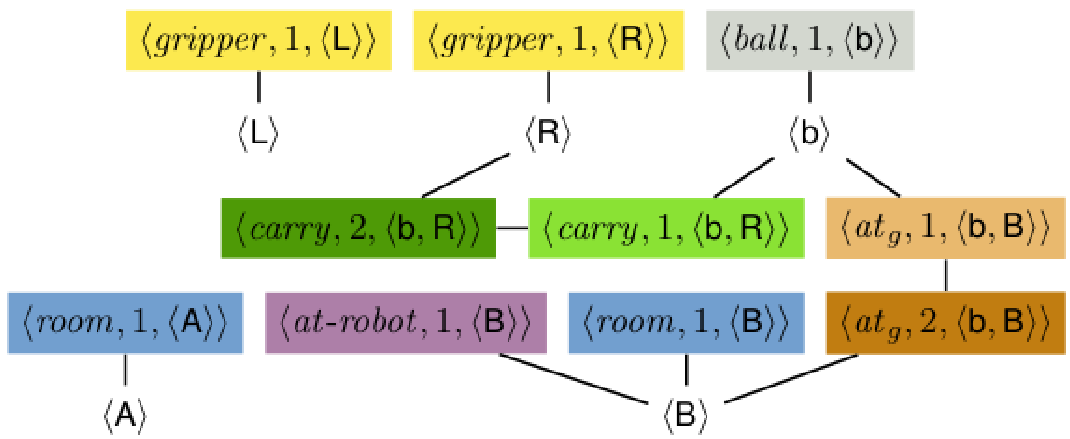
Total number of states in training set ($\mathcal{Q}_{\mathcal{T}}$), and reduced set ($\mathcal{Q}_{\mathcal{T}}/{\sim_{iso}}$)

How state symmetries computed?

States s mapped into vertex-colored graphs $G(s) = (V, E, \lambda)$ with

- **vertices** $v = \langle c \rangle$ with color $\lambda(v) = \perp$ for **constants** c in s,
- **vertices** $v = \langle q, i \rangle$ for all **atoms** $q = p(c_1, \ldots, c_k)$ in s, $i \leq k$, color $\lambda(v) = \langle p, i \rangle$,
- **edges** connect vertices $\langle q, i \rangle$ and $\langle c_i \rangle$ iff $q = p(c_1, \ldots, c_k)$
- **edges** connect vertices $\langle q, i \rangle$ and $\langle q, i+1 \rangle$ iff $q = p(c_1, \ldots, c_k)$ and $i + 1 \leq k$

Theorem: $s \sim_{iso} s'$ iff $G(s) \sim_{iso} G(s')$



State-of-the-art code (`nauty`) to determine if graphs $G(s)$ and $G(s')$ **isomorphic**

Expressivity Requirement of Planning Domains

- General policies see states through **features**; no object or action names
- GNN, 1-WL, and C2 features **don't distinguish** all non-isomorphic states
- 1-WL **distinguishes** s and s', $s \not\sim_{\text{wl}} s'$, if Hist$(G(s)) \neq$ Hist$(G(s'))$
- If 1-WL doesn't distinguish s and s', nor will GNNs or description logics. Let:
  - ▷ **E-conflicts:** $s \not\sim_{\text{wl}} s'$ and $s \not\sim_{iso} s'$
  - ▷ **V-conflicts:** $s \not\sim_{\text{wl}} s'$ and $V(s) \neq V(s')'$ (related to [Horcík and Šír, 2024])
- **V-conflict** implies that GNN can't learn V^* in **training set**
- **E-conflict** implies potential **V-conflict**

Experimental Results: Expressivity Requirements

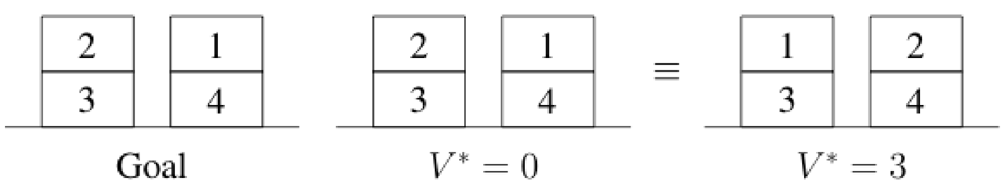
				1-WL		1-WL + G	
Domain	$\#\mathcal{Q}$	$\#\mathcal{S}$	$\#\mathcal{S}/{\sim_{iso}}$	$\#E$	$\#V$	$\#E$	$\#V$
Barman	510	115 M	38 M	1,326	537	1,062	273
Blocks3ops	600	146 K	133 K	50	20	25	0
Blocks4ops	600	122 K	110 K	54	27	27	0
Blocks4ops-clear	120	31 K	3 K	0	0	0	0
Blocks4ops-on	150	31 K	8 K	0	0	0	0
Childsnack	30	58 K	5 K	0	0	0	0
Delivery	540	412 K	62 K	0	0	0	0
Ferry	180	8 K	4 K	36	36	0	0
Grid	1,799	438 K	370 K	42	38	24	20
Gripper	5	1 K	90	0	0	0	0
Hiking	720	44 M	5 M	0	0	0	0
Logistics	720	69 K	38 K	131	131	94	94
Miconic	360	32 K	22 K	0	0	0	0
Reward	240	14 K	11 K	0	0	0	0
Rovers	514	39 M	34 M	0	0	0	0
Satellite	960	14 M	8 M	5,304	4,226	1,708	762
Spanner	270	9 K	4 K	0	0	0	0
Visitall	660	3 M	2 M	0	0	0	0

$\#\mathcal{Q}$ is # number of instances; $\#\mathcal{S}$, $\#\mathcal{S}/{\sim_{iso}}$: # states and partitions. "G" adds predicate $p'(x, y)$ iff $p(x, y)$ and $p_G(x, y)$ true. #E and #V: # of E and V-conflicts.

GNN + RL for General Policies [Ståhlberg *et al.*, 2023]

Domain	Coverage (%)	Domain	Coverage (%)
Blocks	100%	Delivery	100%
Gripper	100%	Miconic	100%
Visitall	100%	Grid	**70%**
Logistics	**36%**	Spanner	**68%**

- Nearly **perfect general policies** obtained in several domains (100%)
- But interesting part is in the **failures**:
  - ▷ **GNN expressivity not enough**
  - ▷ Generality-optimality tradeoff
- Indeed, 1-WL/GNNs can't distinguish pair of states:



Summary

- **Two methods** for learning general policies
  - **Combinatorial:** Explicit pool of features, Min-SAT formulation
  - **GNNs:** Features learned to represent value or policy functions via DRL
- Two **limitations**:
  - **Scalability,** in combinatorial setting
  - **Expressivity** in both settings
- **Computing** symmetries
  - Mapping states into graphs that preserve isomorphisms
  - Using state-of-the-art codes for testing graph isomorphism
- **Results** so far:
  - Savings in combinatorial setting
  - Expressive requirements assessed (not in PRL paper though)
- **Challenge** of obtaining general policies for difficult but tractable domains
  - e.g., N-puzzle, Sokoban (fragments), Pushworld (fragments), etc.

References

[Bajpai et al., 2018] Bajpai, A. N., Garg, S., et al. (2018). Transfer of deep reactive policies for mdp planning. In *Proc. NeurIPS 2018*, pages 10965–10975.

[Barceló et al., 2020] Barceló, P., Kostylev, E., Monet, M., Pérez, J., Reutter, J., and Silva, J.-P. (2020). The logical expressiveness of graph neural networks. In *Proc. ICLR 2020*.

[Drexler et al., 2024] Drexler, D., Ståhlberg, S., Bonet, B., and Geffner, H. (2024). Symmetries and expreessive requirements for learning general policies.

[Fern et al., 2006] Fern, A., Yoon, S., and Givan, R. (2006). Approximate policy iteration with a policy language bias: Solving relational markov decision processes. *JAIR*, 25:75–118.

[Francès et al., 2021] Francès, G., Bonet, B., and Geffner, H. (2021). Learning general planning policies from small examples without supervision. In *Proc. AAAI 2021*, pages 11801–11808.

[Grohe, 2021] Grohe, M. (2021). The logic of graph neural networks. In *Proceedings of the Thirty-Sixth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2021)*, pages 1–17.

[Horcík and Šír, 2024] Horcík, R. and Šír, G. (2024). Expressiveness of graph neural networks in planning domains. In *Proc. ICAPS*.

[Jiménez et al., 2019] Jiménez, S., Segovia-Aguas, J., and Jonsson, A. (2019). A review of generalized planning. *The Knowledge Engineering Review*, 34:e5.

[Khardon, 1999] Khardon, R. (1999). Learning action strategies for planning domains. *AI*, 113:125–148.

[Martín and Geffner, 2000] Martín, M. and Geffner, H. (2000). Learning generalized policies from planning examples using concept languages. In *Proc. KR*.

[Morris et al., 2019] Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. (2019). Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proc. AAAI 2019*, pages 4602–4609.

[Pochter et al., 2011] Pochter, N., Zohar, A., and Rosenschein, J. S. (2011). Exploiting problem symmetries in state-based planners. In *Proc. AAAI 2011*, pages 1004–1009.

[Riddle et al., 2016] Riddle, P., Douglas, J., Barley, M., and Franco, S. (2016). Improving performance by reformulating PDDL into a bagged representation. In *ICAPS 2016 Workshop on Heuristics and Search for Domain-independent Planning*, pages 28–36.

[Rivlin et al., 2020] Rivlin, O., Hazan, T., and Karpas, E. (2020). Generalized planning with deep reinforcement learning. In *ICAPS Workshop on Bridging the Gap Between AI Planning and Reinforcement Learning (PRL)*, pages 16–24.

[Shleyfman et al., 2015] Shleyfman, A., Katz, M., Helmert, M., Sievers, S., and Wehrle, M. (2015). Heuristics and symmetries in classical planning. In *Proc. AAAI 2015*, pages 3371–3377.

[Sievers et al., 2019] Sievers, S., Röger, G., Wehrle, M., and Katz, M. (2019). Theoretical foundations for structural symmetries of lifted PDDL tasks. In *Proc. ICAPS 2019*, pages 446–454.

[Srivastava et al., 2008] Srivastava, S., Immerman, N., and Zilberstein, S. (2008). Learning generalized plans using abstract counting. In *Proc. AAAI 2008*, pages 991–997.

[Ståhlberg et al., 2023] Ståhlberg, S., Bonet, B., and Geffner, H. (2023). Learning general policies with policy gradient methods. In *Proc. KR 2023*.

[Ståhlberg et al., 2024] Ståhlberg, S., Bonet, B., and Geffner, H. (2024). Learning general policies for classical planning domains: Getting beyond c $\_2$. *arXiv preprint arXiv:2403.11734*.

[Toyer et al., 2020] Toyer, S., Thiébaux, S., Trevizan, F., and Xie, L. (2020). ASNets: Deep learning for generalised planning. *Journal of Artificial Intelligence Research*, 68:1–68.